

Figure 1

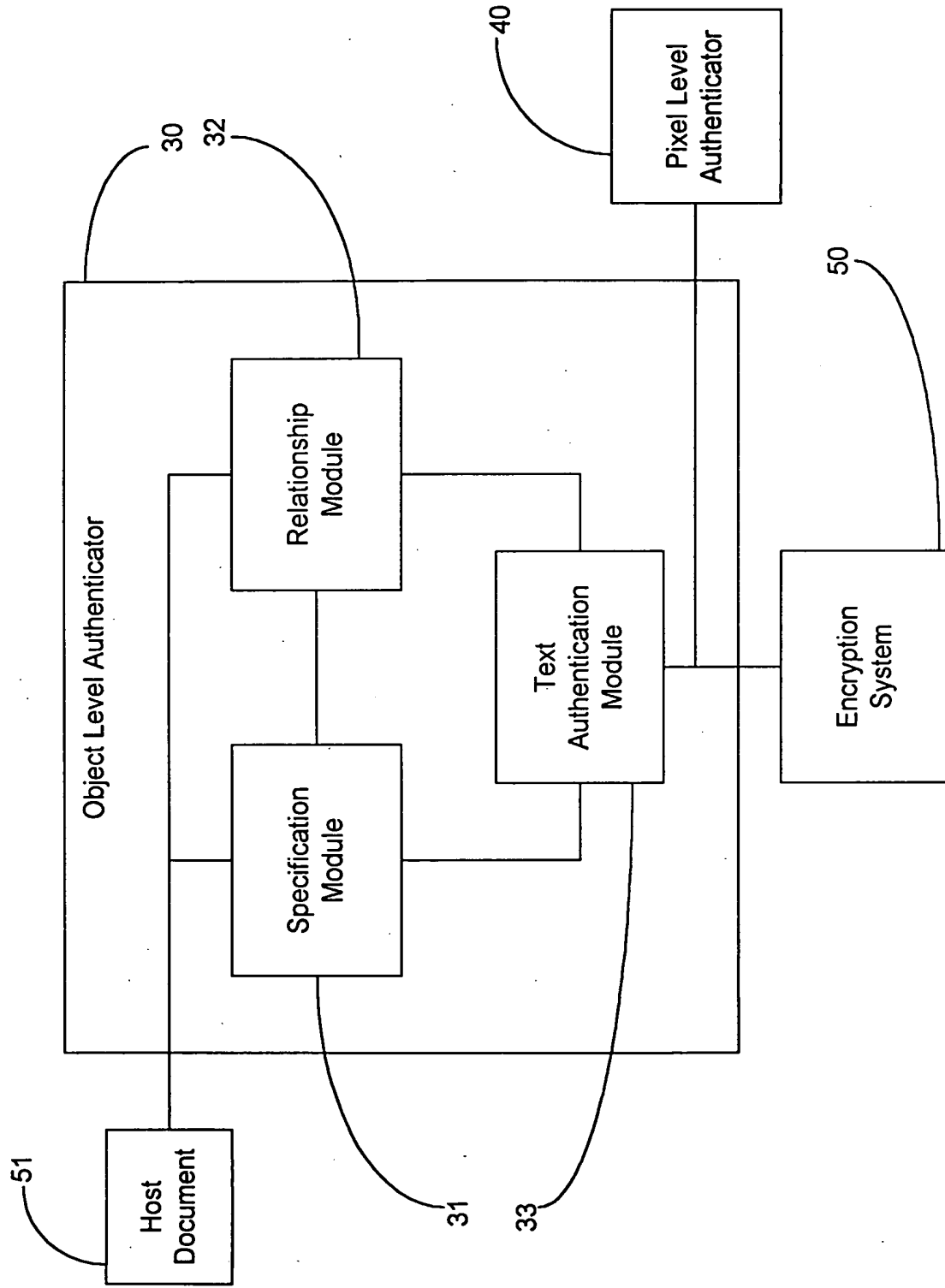


Figure 2

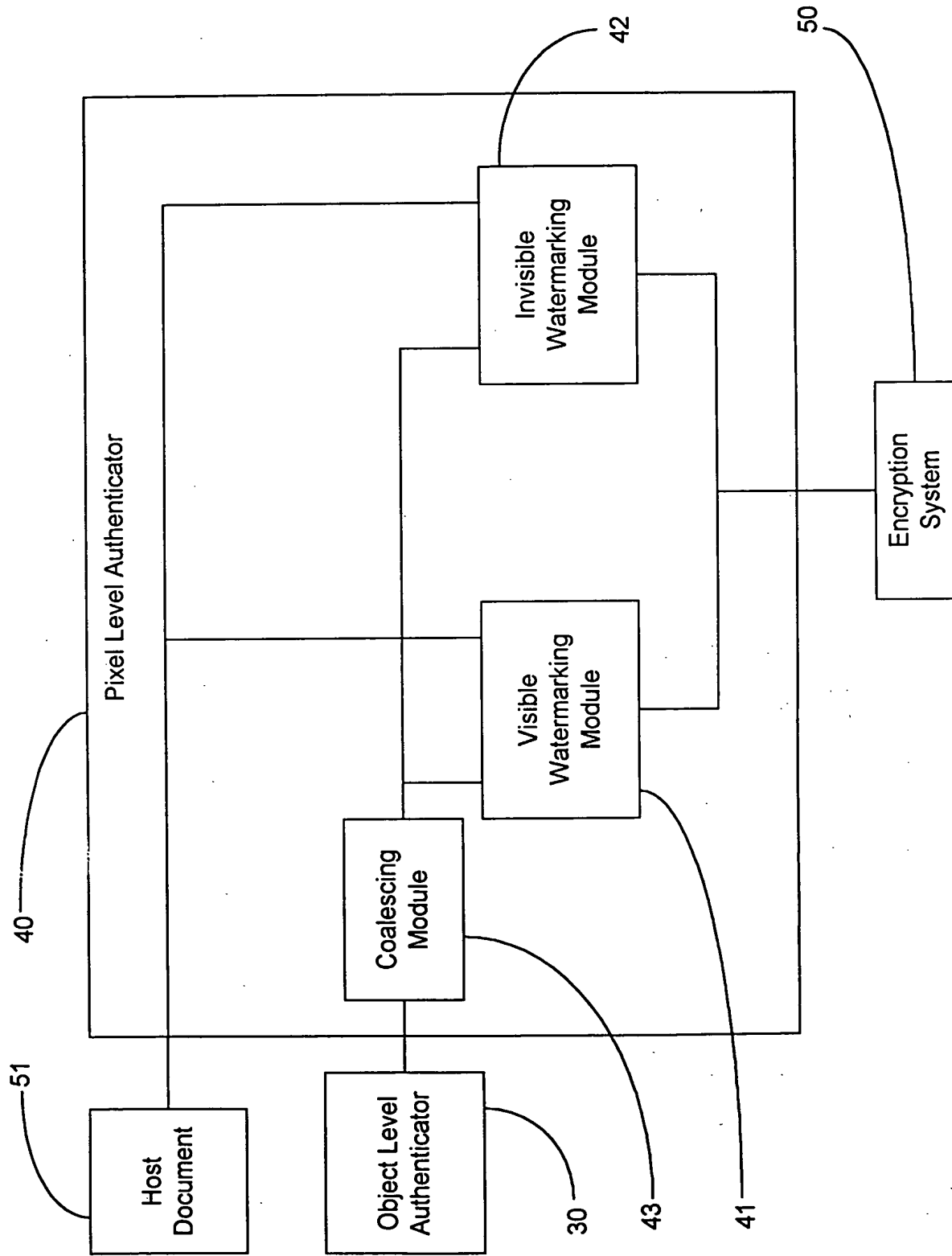


Figure 3

100

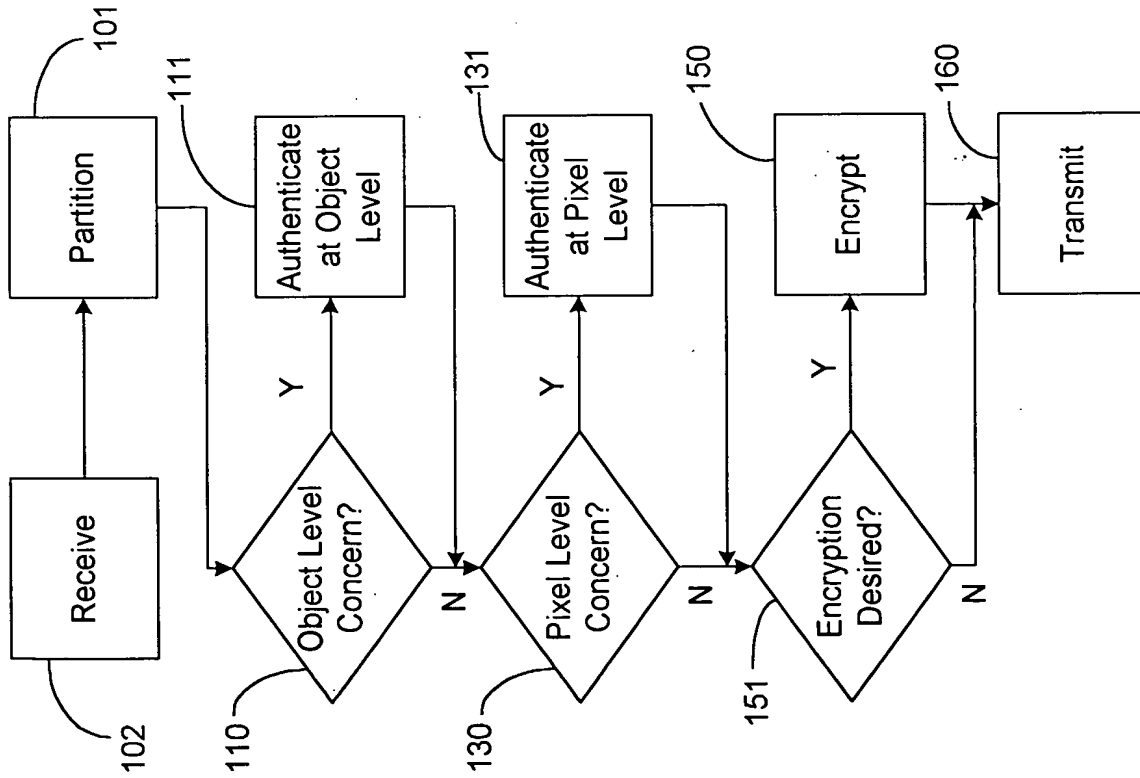


Figure 4

111

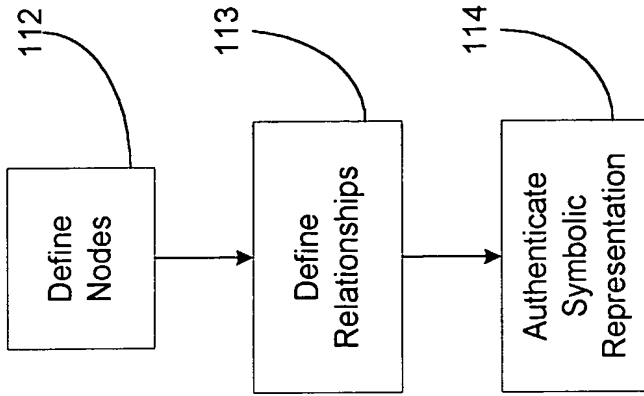


Figure 5

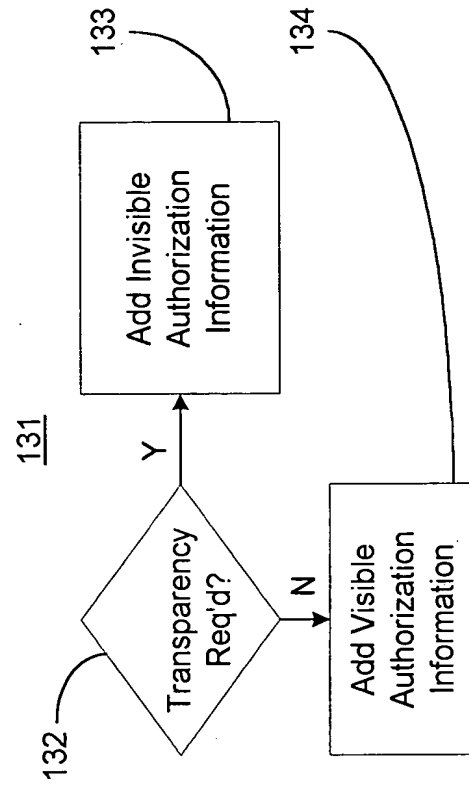


Figure 6

134

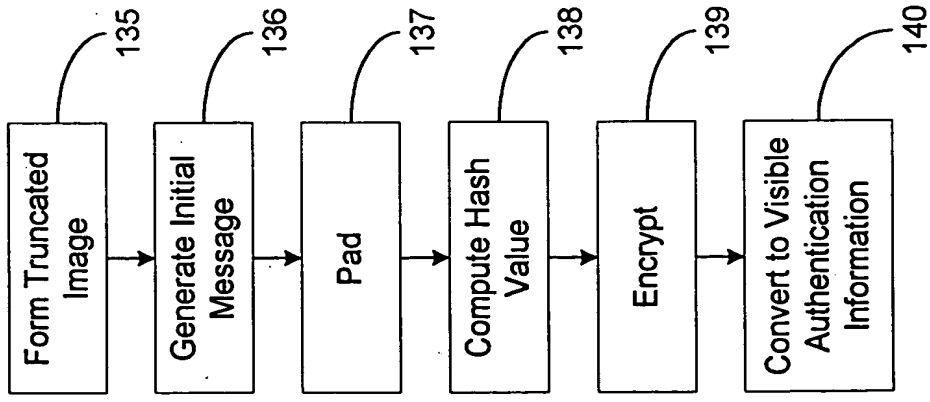


Figure 7

133

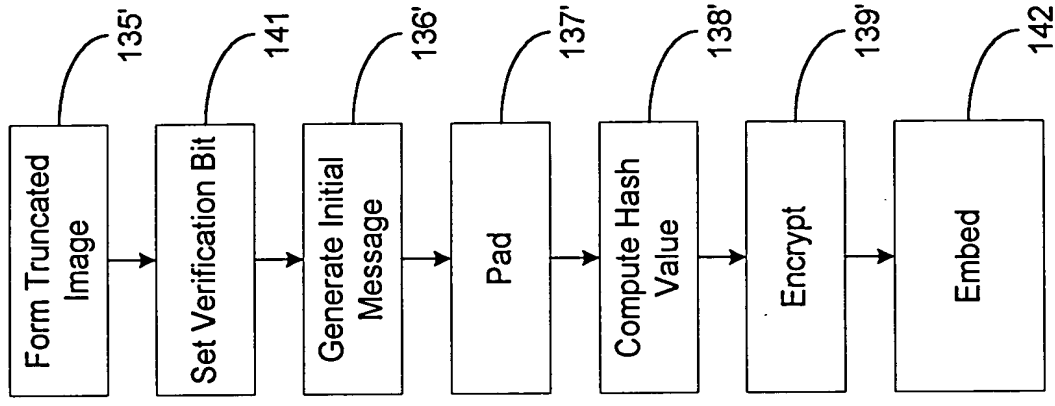


Figure 8

Figure 9

The system flow diagram is illustrated below. It shows the simplicity of the algorithm.

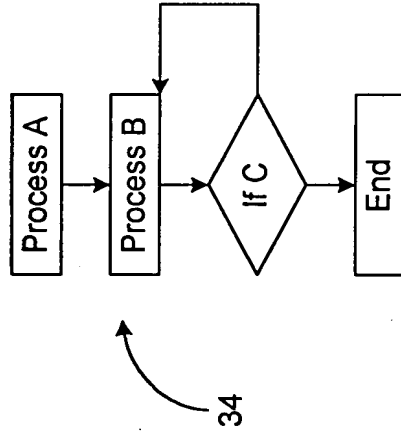


Figure 10

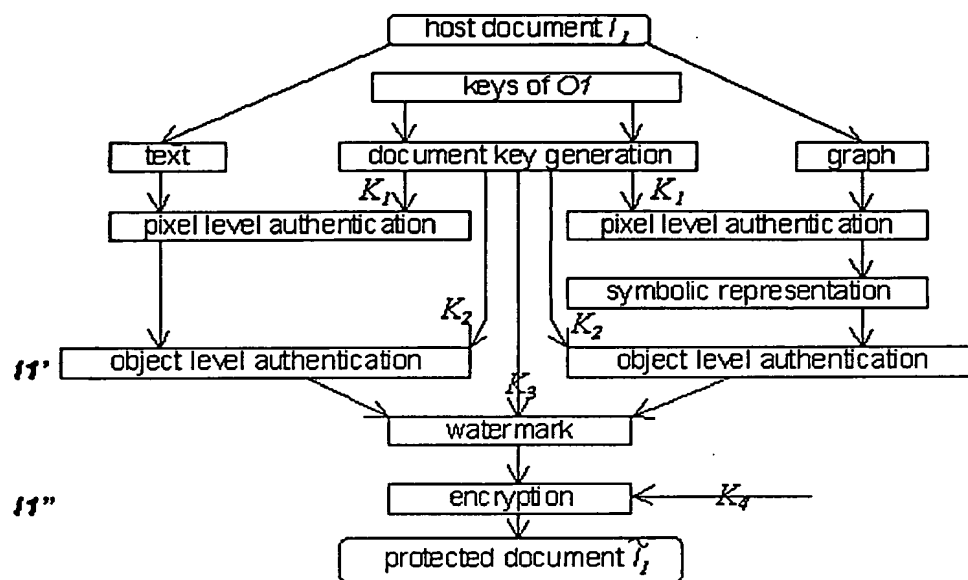


Figure 11



a key set $K = \{K_1, K_2, K_3, K_4\}$

Figure 12

Relationship symbols	
<>	a tuple
C	and
⊃	or
#	not
↑	parent → child
⇄	sibling relation
⇄	twin relation
↓	child ← parent
^	contain relation
—	condition
.	.
.	.
.	.
:	unconnected
Specification symbols	
&	size
#	shape
@	position
©	color

Figure 13

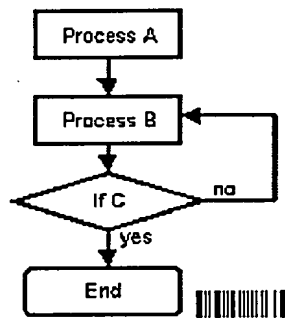


Figure 16

auth
(a) Original size

auth
(b) Enlarged

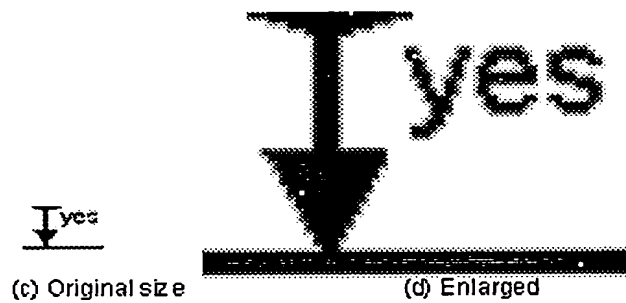
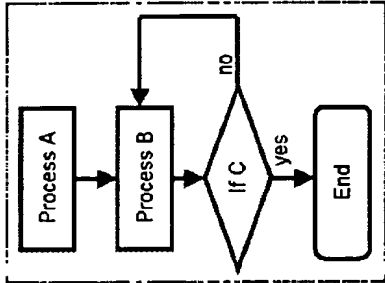


Figure 17

The system flow diagram is illustrated below. It shows the simplicity of the algorithm... "<N1{'Process A', #1, ®, @mid}>→N2{'Process B', #1, ®, @mid}>→N3{'If C', #3, ®, @mid}>→<N4{'End', #2, ®, @mid}>yes; N2|no>>"



34

Figure 14

Figure 15

(Text)	W/o content-dependent one way hash		Our algorithms, w/ content-dependent one way hash	
	Traditional line spacing	Traditional serif length	Coalescing	Object level
Special coding	Needed	Needed	May or may not be needed	May or may not be needed
Imperceptibility	Good	Good	OK	Good
Detectability	Bad	Bad	OK	Good
Pixel-level detectability	Bad	Bad	Good if Method I OK if Method II	Can't detect
Localization-ability	Bad	Some bad. Some OK	OK	Good
Copy and print	Bad	Bad	Good if Method I, bad if Method II	Good
Noise resistance-ability	Bad	OK	Good if Method I, bad if Method II	Good
Robustness to scaling	Good	OK	OK if Method I, bad if Method II	Good

Figure 18